Deciding weak weighted bisimulation

Marino Miculan[1⋆] and Marco Peressotti[2⋆⋆]

[1] Department of Mathematics, Computer Science and Physics, University of Udine
[2] Department of Mathematics and Computer Science, University of Southern Denmark

Abstract Weighted labelled transition systems are LTSs whose transitions are given weights drawn from a commutative monoid, subsuming a wide range of systems with quantitative aspects. In this paper we extend this theory towards other behavioural equivalences, by considering *semirings* of weights. Taking advantage of this extra structure, we consider a general notion of *weak weighted bisimulation*, which coincides with the usual weak bisimulations in the cases of non-deterministic and fully-probabilistic systems. We present a general algorithm for deciding weak weighted bisimulation. The procedure relies on certain systems of linear equations over the semiring of weights hence it can be readily instantiated to a wide range of models. We prove that these systems admit a unique solution for ω-continuous semirings.

1 Introduction

Many extensions of labelled transition systems have been proposed for dealing with quantitative notions such as execution times, probabilities and stochastic rates; see *e.g.* [6, 12, 13, 16, 29] among others. This plethora of variants has naturally pointed out the need for general theories and tools, covering uniformly a wide range of cases. As examples of these theories we mention *state-to-function transition systems* (FuTSs) [10, 19], *uniform labelled transition systems* (ULTraSs) [7, 22, 24] and *weighted labelled transition systems* (WLTSs) [15, 17]. In particular, in a WLTSs every transition is associated with a *weight* drawn from a commutative monoid; the monoid structure defines how weights of alternative transitions combine. As we will recall in Section 2, by suitably choosing this monoid we can recover ordinary non-deterministic, probabilistic, and stochastic transition systems, among others. The WLTS meta-model offer a notion of *(strong) weighted bisimulation*, which can be readily instantiated to particular cases obtaining precisely the well-known Milner's strong bisimulation [26], Larsen and Skou's strong probabilistic bisimulation [18], strong stochastic bisimulation [13], *etc.*

However, in many situations strong bisimulations are too fine, and many other relations have been introduced. Likely the most widely known of these equivalences is Milner's *weak bisimulation* for LTSs [26] (see [30] for others). Weak bisimulations

⋆ Partially supported by PRID 2017 *ENCASE*. marino.miculan@uniud.it
⋆⋆ Supported by the CRC project, grant no. DFF–4005-00304 from the Danish Council for Independent Research, and by the Open Data Framework project at the University of Southern Denmark. peressotti@imada.sdu.dk

focus on systems' interactions (communications, synchronizations, *etc.*), ignoring transitions associated with systems' internal operations, hence called *silent* or *unobservable* (and denoted by the τ). Although, τ-labelled transitions cannot be directly observed, their effects (delays, probability distributions) are still observable and hence cannot be ignored. Notions of weak bisimulations for systems of interests have recently been proven instances of a uniform and abstract notion of coalgebraic weak bisimulation [8, 9]. In order to instantiate this definition to WLTSs and obtain *weak weighted bisimulation* in a principled way, we consider systems weighted over a *semiring*: roughly speaking, the added multiplicative structure on weights allows us to compositionally extend weights to multi-step transitions and traces. In Section 3 we show that the resulting notion of weak bisimulation coincides with the known ones in the cases of non-deterministic and fully probabilistic systems, just by changing the underlying semiring.

Then, in Section 4 we present the general algorithm for computing weak weighted bisimulation equivalence classes, parametric in the underlying semiring. This algorithm is a variant of Kanellakis and Smolka's algorithm for deciding strong non-deterministic bisimulation [14]. Our solution builds on the refinement technique used for the *coarsest stable partition*, but instead of "strong" transitions in the original system we consider "weakened" ones. We prove that this algorithm is correct, provided the semiring satisfies some mild conditions, *i.e.* it is ω-complete and ω-continuous. Finally, we discuss also its complexity, which is comparable with Kanellakis and Smolka's algorithm. Thus, this algorithm can be used in the verification of many kinds of systems, just by replacing the underlying semiring (boolean, probabilistic, stochastic, tropical, arctic, ...) and taking advantage of existing software packages for linear algebras over semirings.

Some final remarks and directions for further work are in Section 5. Additional examples can be found in the companion report [21].

2 Weighted labelled transition systems

This section recalls the notion of *labelled transition systems weighted over a commutative monoid* (WLTS) and their (strong) bisimulation called *weighted bisimulation* [15].

A commutative (a.k.a. abelian) monoid is a set M equipped with a binary operation $+$ that is associative, commutative and has a unit 0. In the sequel let $(M, +, 0)$ be any commutative monoid. As usual, a monoidal structure will be often denoted by its set, when the intended structure is clear from the context.

Definition 1 (M-WLTS [15, Def. 1]). *A* M-weighted labelled transition system *is a triple* (X, Σ, ρ) *where:*

- X *is a set of* states *(processes);*
- Σ *is an at most countable set of* labels*;*
- $\rho\colon X \times \Sigma \times X \to M$ *is a* weight function*, mapping each triple of* $X \times \Sigma \times X$ *to a weight drawn from* M*.*

(X, Σ, ρ) *is said to be* image finite *(resp.* countable*) whenever for each* $x \in X$ *and* $\sigma \in \Sigma$*, the* $\{y \in X \mid \rho(x, \sigma, y) \neq 0\}$ *is finite (resp. countable).*

The monoidal structure was not used in Definition 1 except for the existence of a distinguished element required by the image finiteness (resp. countability) property. The commutative monoidal structure of weights comes into play in the notion of *weighted bisimulation*, where weights of transitions to any state in a given equivalence class are be combined[3]. This operation is commonplace for stochastic, probabilistic, and rated transition systems, but at first it may appear confusing with respect to the notion of bisimulation of non-deterministic systems.

Definition 2 (Strong weighted bisimulation [15, Def. 3]). *Given a M-WLTS (X, Σ, ρ), a* (strong) M-weighted bisimulation *is an equivalence relation R on X such that for each pair (x, x') of elements of X, $(x, x') \in R$ implies that for each label $\sigma \in \Sigma$ and each equivalence class C of R:*

$$\sum_{y \in C} \rho(x, \sigma, y) = \sum_{y \in C} \rho(x', \sigma, y). \tag{1}$$

Processes x and x' are said to be M-bisimilar *(or just bisimilar when M is understood) if there exists a M-bisimulation $\sim_M$ such that $x \sim_M x'$.*

Weighted bisimulations are closed under arbitrary unions. As a consequence, weighted bisimilarity on any M-WLTS is the largest M-bisimulation over it.

Observe that Definition 2 is well-given only if the summations (1) are defined. Weighted bisimulation is always defined on image-finite systems and it is defined on image-countable ones provided the underlying weighting structure supports sums over countable families. In fact, [15] restricts to image finite systems (which is not unusual in the coalgebraic setting). This paper considers also image-countable systems since this generalization will be crucial to the definition of weak and delay bisimulations: the unfolding of *tau*-loops yields countably many executions even if the original system is image-finite.

Monoids with countable sums are often called *ω-complete*[4] and in practice requiring M to be ω-complete is not a severe restriction, since the commutative monoids relevant for most systems of interest admit summations over countable sets. To support this claim, the rest of this section illustrates how several systems of interests are indeed instances of WLTS over commutative ω-monoids.

Example 1 (Non-deterministic systems). Consider the commutative (ω-complete) monoid of logical values equipped with disjunction $\mathbb{B} \triangleq (\{\mathtt{tt}, \mathtt{ff}\}, \vee, \mathtt{ff})$. $\mathbb{B}$-WLTSs correspond to non-deterministic LTSs and $\mathbb{B}$-bisimulations to classical bisimulation equivalences. In fact, every $\mathbb{B}$-valued weight function is an appurtenance predicate defining a subset of its domain rendering $\rho : X \times \Sigma \times X \to \mathbb{B}$ equivalent to the classical definition of the transition relation $\to \subseteq X \times \Sigma \times X$ for LTSs. Finally, taking logical disjunction as the summation in Definition 2 encodes the ability to reach C making an a-labelled transition.

[3] The associative and commutative properties reflect the fact that equivalence classes are sets and transitions to them do not come in any predefined order.

[4] Monoids can be readily extended to ω-monoids adding either colimits freely or an "∞" element. Likewise, ω-completeness can be relaxed w.r.t. a given σ-algebra by considering σ-semirings.

Example 2 (Rated systems). Let $(\mathbb{R}_0^+, +, 0)$ be the monoid of non-negative real numbers with addition. $\mathbb{R}_0^+$-WLTSs corresponds to rated transition systems [16, 17] (which model stochastic systems) and $\mathbb{R}_0^+$-bisimilarity is rated bisimilarity (strong equivalence [13]). Although $\mathbb{R}_0^+$ is not ω-complete, its completion is the monoid of non-negative reals extended with positive infinity $\overline{\mathbb{R}}_0^+$. Systems weighted over $\mathbb{R}_0^+$ are recovered as systems over $\overline{\mathbb{R}}_0^+$ and their bisimulations as well.

Example 3 (Probabilistic systems). Probabilistic systems are a particular case of $\mathbb{R}_0^+$-WLTSs where weight functions are required to satisfy suitable constraints. In particular, a $\mathbb{R}_0^+$-WLTSs (X, Σ, ρ) is a fully-probabilistic system for any state $x \in X$ it holds that $\sum_{\sigma\in\Sigma, y\in X} \rho(x, \sigma, y) \in \{0, 1\}$. Then, probabilistic bisimulations coincide with $\mathbb{R}_0^+$-bisimulations [16].

3 Weak weighted bisimulation

Weak bisimulation weakens strong bisimulation by allowing sequences of unobservable actions (denoted by the distinguished label $\tau \notin \Sigma$) before and after any observable one (any $\sigma \in \Sigma$). Reworded, what to an external observer appears as a single labelled (weighted) transition arises instead from the contribution of several transitions. The operation governing which and how transitions are combined and presented to the observer is called (parameterised) *saturation* [9].

In the case of weighted transition systems, saturation requires weights to be equipped with some additional structure: that of an ω-continuous semiring. A semiring is a set S equipped with a commutative monoid structure $(S, +, 0)$ (called addition) and a monoid structure $(S, \cdot, 1)$ (called multiplication) such that $+$ distributes over the $\cdot$ and 0 annihilates $\cdot$. A semiring $(S, +, 0, \cdot, 1)$ is called *positively ordered* whenever its carrier S admits a partial order $(S, \leq)$ such that the unit 0 is the bottom element of this ordering and semiring operations are monotonic in both components *i.e.* if it holds that $a \leq b$ implies that $a \diamond c \leq b \diamond c$ and $c \diamond a \leq c \diamond b$ for $\diamond \in \{+, \cdot\}$ and any $a, b, c \in S$. A semiring is positively ordered if and only if it is *zerosumfree i.e.* if it holds that $a = b = 0$ whenever $a+b = 0$. The natural order $a \lhd b \iff \exists c. a + c = b$ is the weakest order that render S positively ordered. A positively ordered semiring is said to be ω-*complete* if it has countable sums given on any countable family as $\sum_{i<\omega} a_i = \sup\{\sum_{j\in J} a_j \mid J \subset \omega\}$. It is called ω-*continuous* if suprema of ascending ω-chains exist and are preserved by both operations *i.e.*: $a \diamond \bigvee_{i<\omega} b_i = \bigvee_i a \diamond b_i$ and $\bigvee_i b_i \diamond <\omega\ a = \bigvee_i b_i \diamond a$ for $\diamond \in \{+, \cdot\}$. Examples of such semirings are: the boolean semiring, the arithmetic semiring of non-negative real numbers with infinity and the tropical semiring.

Henceforth let $(S, +, 0, \cdot, 1)$ be an ω-complete and ω-continuous semiring. As for monoids, semirings will be often denoted by their carrier provided the structure is clear from the context.

As before, the addition structure weights branching computations as the sum of each branch weight. Likewise, multiplication weights sequences in a modular fashion given the weight of each step. Distributivity and annihilation guarantee coherence of the two operations: weighting computations by multiplying along

sequences and then summing branches or *vice versa* is equivalent. For instance, probabilities are multiplied along sequences for steps are dependent events and are summed along branches for each branch is an independent event. The order structure is required by the notion of saturation which relies on binary joins to merge some possibly overlapping contributions and partial executions in a coherent way. For instance, the weight of computations arising from transitions loops are obtained as the sumpremum of all weights of their finite unfoldings. Here, ω-continuity guarantees that this approach is always possible and that it is coherent with the rest of the weighting structure.

For an alphabet Σ let $\Sigma_\tau \triangleq \Sigma + \{\tau\}$ where τ is the label denoting unobservable transitions. Transition weights drawn from an ω-complete and ω-continuous semiring S. All WLTSs and weight functions are assumed image countable. The notion of parametrised saturation instantiate to WLTSs as follows.

Definition 3. *The* saturation *of* $\rho\colon X \times \Sigma_\tau \times X \to S$ *w.r.t. the parameter* $h\colon X \to Y$ *is the function* $\rho^*_h\colon X \times \Sigma_\tau \times Y \to S$ *corresponding to the least solution to the equation system:*

$$\rho^*_h(x,\tau,y) \triangleq (\delta_y \circ h)(x) \vee \sum_{x' \in X} \rho(x,\tau,x') \cdot \rho^*_h(x',\tau,y)$$

$$\rho^*_h(x,\sigma,y) \triangleq \sum_{x' \in X} \rho(x,\sigma,x') \cdot \rho^*_h(x',\tau,y) + \sum_{x' \in X} \rho(x,\tau,x') \cdot \rho^*_h(x',\sigma,y)$$

where $\delta_y\colon Y \to [0,1]$ *is Dirac's delta function on* y*:*

$$\delta_y(y') = \begin{cases} 1 & \text{if } y = y' \\ 0 & \text{otherwise.} \end{cases}$$

Roughly speaking, the parameter h describes a quotient of the system state space X and will be later used to represent equivalence classes in the definition of weighted bisimulation. In this sense, the function ρ^*_h weights computations from states to equivalence classes as perceived by an external observer. The first family of equations of the system defines the weight of saturated unobservable transitions taking x to y by recursion on computations starting at x with a τ-labelled step (right term of the join) combined with loops inside the class of x as per h (left term). Observe that loops are given weight 1 since they model the possibility of (unobservable) computations internal to the class of x and must not affect computations leaving the class of x. The second family of equations defines the weight of saturated observable transitions by recursion on computations starting at x with an observable step (left term) or an τ-labelled step (right term). In the first case, the recursion continues considering only unobservable saturated transitions (the only observable has already be consumed) whereas in the second the recursion proceeds considering only observable saturated transitions (the observable has not been consumed yet). Finally, note that parameterised saturation is always defined on semirings that are ω-complete and ω-continuous.

Recall that strong weighted bisimulations relate states whose transition to each equivalence class have the same combined weight. Weak bisimulations are

defined likewise except that saturated transitions are considered in order to abstract from unobservable moves. The saturation parameter is the quotient induced by the weak bisimulation equivalence since we are interested in the combined weight assigned to reaching each equivalence class.

Definition 4 (Weak weighted bisimulation [9, Sec. 4.3]). *An equivalence relation R on X is a* weak (weighted) bisimulation *whenever $x\,R\,x'$ implies that for any equivalence class $C \in X/R$ and label $\sigma \in \Sigma + \{\tau\}$:*

$$\rho^*_\kappa(x, \sigma, C) = \rho^*_\kappa(x', \sigma, C)$$

where $\kappa\colon X \to X/R$ is the canonical projection taking each element of X to its equivalence class w.r.t. R.

The remaining of the section illustrates some examples of weak weighted bisimulation and possible applications. The interested reader is referred to [9, 21] for more detailed examples.

Non-determinism Non-deterministic LTSs are $\mathbb{B}$-WLTSs where $\mathbb{B}$ denotes the *boolean semiring* $(\{\mathtt{tt}, \mathtt{ff}\}, \vee, \mathtt{ff}, \wedge, \mathtt{tt})$. It is shown in [9, Prop. 4.2] that the notions of weak non-deterministic bisimulation as per [26] and that of weak $\mathbb{B}$-weighted bisimulation coincide.

Probabilities Probabilistic systems are recovered as a special case of systems weighted over the semiring of non-negative real numbers extended with infinity. The notions of weak probabilistic bisimulation as per [4] and weak weighted bisimulation coincide on fully probabilistic systems as shown in [9, Prop. 4.11]. Another semiring used for reasoning about probabilistic events is $([0, 1], \max, 0, \cdot, 1)$. This is used to model the maximum likelihood of events, *e.g.* for troubleshooting, diagnosis, failure forecasts, worse cases, etc. A weak bisimulation on this semiring allows to abstract from "unlikely" events, focusing on the most likely ones.

Resources There are several semirings that can be used for reasoning about different aspects of resource dependant computations. The first example is the family of arithmetic semirings whose addition substructure is also used in the definition of rated systems (*cf.* Example 2). A prominent example are systems weighted over the free completion of natural numbers $\mathbb{N}+\{\infty\}$. Transition weights model resource consumption and weak weighted bisimulation corresponds to *resource bisimulation* from [1] as shown in [9, Cor. 4.10].

Other examples are the tropical semiring, its dual the arctic semiring, and their generalisations. Some instances are $(\overline{\mathbb{R}}, \min, +\infty, +, 0)$; $(\overline{\mathbb{R}}, \max, -\infty, +, 0)$; $(\overline{\mathbb{R}}, \min, +\infty, \max, -\infty)$. These semirings are used to model optimisation problems such as task scheduling and routing. In these contexts, weak bisimulation would allow to abstract from "unobservable" tasks *e.g.* internal tasks and treat a cluster of machines as a single one. Similar to the last family is that of truncation semirings like $(\{0, \ldots, k\}, \max, 0, \min\{\_ + \_, k\}, k)$. This variant supports reasoning up to a given threshold k. Weak weighted bisimulation on truncation semirings allows *e.g.* reasoning about if and when resources consumption violates the threshold but not about how much.

4 Deciding weak weighted bisimulation

The section describes an algorithm for computing the coarsest weak weighted bisimulation. The algorithm is parametric in the semiring structure of weights hence can be used in the mechanized verification and analysis of many kinds of systems. Algorithms like this are often called "universal" in the sense that they do not depend on any particular numerical domain nor its machine representation [20].

The algorithm is a variation of Kanellakis and Smolka's algorithm for deciding strong non-deterministic bisimulation [14] and builds on the refinement technique used for the *coarsest stable partition*, but instead of "strong" transitions in the original system saturated ones are considered during the splitting phase. The idea of deciding weak bisimulation by computing the strong bisimulation equivalence classes for the saturated version of the system has been previously and successfully used *e.g.* for non-deterministic or probabilistic weak bisimulations [3]. The resulting complexity is basically that of the coarsest stable partition problem plus that introduced by the construction of the saturated transitions. The last factor depends on the properties and kind of the system and, in our case, on the properties of the semiring M. In particular note that, differently from the non-deterministic case, saturated transitions will be computed on-demand since without the assumptions at the base of the double-arrow construction, pre-computing them would be exponential in the number of states.

We remark that solving linear equations over arbitrary semirings is out of the scope of this work and that parametrisation henceforth is assumed to provide a procedure for computing saturated transitions.

Before outlining the general idea of the algorithm let us introduce some notation. For a finite set X we denote by $\mathcal{X}$ a partition of it *i.e.* a set of pairwise disjoint sets $X = \biguplus \mathcal{X} = \biguplus\{B_0, \ldots, B_n\}$. We shall refer to the elements of the partition $\mathcal{X}$ as *blocks* or *classes* since every partition induces an equivalence relation $\bigcup_{B\in\mathcal{X}} B \times B$ on X and vice versa.

Given a finite WLTS (X, Σ_τ, ρ) the general idea for deciding weak weighted bisimulation by partition refinement is to start with a partition of the states $\mathcal{X}_0$ coarser than the weak bisimilarity relation *e.g.* $\{X\}$ and then successively refine the partition with the help of a *splitter* (*i.e.* a witness that the partition is not stable). This process eventually yields a partition $\mathcal{X}_k$ being the set of equivalence classes of the weak bisimilarity. A splitter of a partition $\mathcal{X}$ is a pair made of an action and a class of $\mathcal{X}$ that violates the condition for $\mathcal{X}$ to be a weak bisimulation. Reworded, a pair $\langle \sigma, C\rangle \in \Sigma_\tau \times \mathcal{X}$ is a splitter for $\mathcal{X}$ if, and only if, there exist $B \in \mathcal{X}$ and $x, y \in B$ such that:

$$\rho_h^*(x, \sigma, C) \neq \rho_h^*(y, \sigma, C) \tag{2}$$

where the parameter $h\colon X \to \mathcal{X}$ is the quotient induced by the current partition $\mathcal{X}$. Then, $\mathcal{X}_{i+1}$ is obtained from $\mathcal{X}_i$ by splitting every[5] $B \in \mathcal{X}_i$ accordingly to the

[5] In Kanellakis and Smolka's algorithm, only the block B is split but in our case we need to evaluate every block anyway because of on-the-fly saturation.

```
1: 𝒳 ← {X}
2: 𝒳′ ← ∅
3: repeat
4:    changed ← false
5:    𝒳″ ← 𝒳
6:    for all C ∈ 𝒳 \ 𝒳′ do
7:       for all σ ∈ Σ + {τ} do
8:          if ⟨σ, C⟩ is a splitter then
9:             𝒳 ← ⋃{B/≈_{σ,C} | B ∈ 𝒳}
10:            changed ← true
11:         end if
12:      end for
13:   end for
14:   𝒳′ ← 𝒳″
15: until not changed
16: return 𝒳
```

Figure 1. The algorithm for weak weighted bisimulation.

selected splitter $\langle \sigma, C \rangle$.

$$\mathcal{X}_{i+1} \triangleq \bigcup \left\{ B/\underset{\sigma,C}{\approx} \mid B \in \mathcal{X}_i \right\} \tag{3}$$

where $\underset{\sigma,C}{\approx}$ is the equivalence relation on states induced by the splitter *i.e.*:

$$x \underset{\sigma,C}{\approx} y \overset{\triangle}{\iff} \rho_h^*(x,\sigma,C) = \rho_h^*(y,\sigma,C).$$

Note that differently from the case of LTSs, blocks can split in more than two parts since weights of outgoing transitions may be more than two.

This procedure is implemented by the pseudocode in Figure 1. Given a finite WLTS (X, Σ_τ, ρ) as input, it returns a partition $\mathcal{X}$ of X induced by a weak weighted bisimulation for the system. Observe that initialising the partition as $\{X\}$ yields weak weighted bisimilarity *i.e.* the greatest weak bisimulation. Other bisimulations can be obtained from different initialisations.

Starting from $\{X\}$, the partition $\mathcal{X}$ takes every value of the sequence $\mathcal{X}_k$ where new values are computed applying the splitting step described above to the current value of $\mathcal{X}$ until a fixed point is reached. Besides the current partitions $\mathcal{X}$, two auxiliary partitions ($\mathcal{X}'$ and $\mathcal{X}''$) are required in order to track which classes are added to $\mathcal{X}$ during the previous iteration of the repeat-until loop and thus avoid checking splitters already used. These additional partitions improve readability but the same result can be achieved, for instance, labelling previously checked blocks. Moreover, $\mathcal{X}'$ and $\mathcal{X}''$ make the flag *changed* redundant. The algorithm iterates over each candidate splitter $\langle \sigma, C \rangle$ and tries to split the partition by checking whether (2) holds. If the partition "survives" every splitting test then it is stable and in particular it describes a weak M-bisimulation. Saturated transitions required by each split test are obtained as solutions to the linear

```
1: 𝒳 ← {X}
2: 𝒳′ ← ∅
3: repeat
4:    changed ← false
5:    for all C ∈ 𝒳 \ 𝒳′ do
6:       for all σ ∈ Σ + {τ} do
7:          compute and sort ρ*_h(x, σ, C) by block and weight
8:       end for
9:       if there is a splitter then
10:         𝒳′ ← 𝒳
11:         𝒳 ← refine(𝒳, C)
12:         changed ← true
13:      end if
14:   end for
15: until not changed
16: return 𝒳
```

Figure 2. An alternative algorithm for linearly ordered blocks and weights.

equations system from Definition 3. For each equivalence class tested C, $|\Sigma| + 1$ systems of $|X|$ equations and unknowns are solved but equations associated with the τ occur every system and hence can be reused.

The complexity of solving these systems depends on the underling semiring structure. For instance, solving a system over the semiring of non-negative real numbers is in P [2], whereas solving a system over the tropical (resp. arctic) semiring is in NP∩coNP [11]. Since the algorithm is parametrized by the semiring, its complexity will be parametrized by the one introduced by the solution of these linear equation systems. Therefore we shall denote by $\mathcal{L}_S(n)$ the complexity of solving a system of n equations in n variables over S.

Let n and m denote the cardinality of states and labels respectively. For each block C used to generate splits, there are exactly m candidates requiring to solve m split tests and perform at most m updates to $\mathcal{X}$. Splits can be thought describing a tree whose nodes are the various blocks encountered by the algorithm during its execution and whose leaves are exactly the elements of the final partition. Because the cardinality of $\mathcal{X}$ is bound by n, the algorithm can encounter at most $\mathcal{O}(n)$ blocks during its entire execution and hence it performs at most $\mathcal{O}(n)$ updates of $\mathcal{X}$ (which happens when splits describe a perfect tree with n leaves). Therefore, in the worst case, the algorithm does $\mathcal{O}(nm)$ split tests and $\mathcal{O}(n)$ partition refinements. Partition refinements and checks of (2) can be both done in $\mathcal{O}(n^2)$ without any additional assumption about X, A and M nor the use of particular data structures or primitives. Therefore the asymptotic upper bound for time complexity of the proposed algorithm is $\mathcal{O}(nm(\mathcal{L}_S(n)+n^2))$ where $\mathcal{L}_S(n)$ is the upper bound for the complexity introduced by computing the weak transitions for a given set of states.

The order structure on weights assumed by Definition 3 can be exploited to speed-up the search for splitters. Under the mild assumption that weights and blocks are totally ordered (at least within the same partition) saturated

transition to a splitting block C can be lexicographically ordered. Then, a block B with more than one entry for the same label, say σ, is split by $\langle \sigma, C \rangle$: the presence of different entries is due to corresponds to two states x and y in B such that (2) holds. The resulting algorithm is reported in Figure 2. For each $\langle \sigma, C \rangle$ there are at most n weak transitions $\rho_h^*(x, \sigma, C)$ and these are sorted in $\mathcal{O}(n\ln(n))$—or in $\mathcal{O}(n)$ using a classical algorithm from [2]. On the worst case the algorithm encounters $\mathcal{O}(n)$ blocks during its entire execution yielding a worst case time complexity in $\mathcal{O}(nm(\mathcal{L}_S(n) + n))$.

Theorem 1. *The asymptotic upper bound for time complexity of the algorithm is in* $\mathcal{O}(nm(\mathcal{L}_S(n) + n^2))$*, and in* $\mathcal{O}(nm(\mathcal{L}_S(n) + n))$ *given a linear ordering for blocks and weights. Both algorithms have space complexity in* $\mathcal{O}(mn^2)$*.*

5 Concluding remarks

In this paper we have introduced a general algorithm for deciding weak weighted bisimulation parametric in the underlying algebraic structure of weights. The algorithm generalises Kanellakis and Smolka's algorithm for the coarsest stable partition problem to weighted saturated graphs. The core of the algorithm is the definition of the splitting step in terms of the saturated system which is computed on-the-fly as necessary. The algorithm is parametric in the underlying structure of weights and can be readily instantiated to many systems of interest.

Obviously, for specific systems and semirings there are *ad hoc* algorithms more efficient than ours. For instance, situations where saturation is invariant under partition changes allow for more efficient approaches that pre-compute the saturated systems (like, e.g., the transitive closure in nondeterministic LTSs). However, in general this is not always possible; a prominent example are probabilistic and stochastic systems. In the case of systems whose weights are drawn from the semiring of non-negative real numbers (which captures probabilistic systems) the asymptotic upper bound for time complexity of our algorithm is $\mathcal{O}(mn^{3.8})$ (since $\mathcal{L}_{\mathbb{R}_0^+}(n)$ is in $\mathcal{O}(n^{2.8})$ using [2]). However, deciding weak bisimulation for fully-probabilistic systems is in $\mathcal{O}(mn^3)$ on the worst case using the algorithm by Baier et al. [5] (the original analysis fixes Σ resulting in the worst case complexity $\mathcal{O}(n^3)$). Their algorithm leverages properties not available at the general level of WLTSs (even under the assumption of suitable orderings), such as: sums of outgoing transitions are bounded, there are complementary events, real numbers have more structure than a semiring, weak and delay bisimulations coincide for finite fully-probabilistic systems (*e.g.* this does not hold for non-deterministic LTSs). A future work is to generalize the efficient results of [5] to a parametrized algorithm valid for a class of WLTSs subject to suitable constraints.

Another direction for improving the efficiency of our algorithm is to extend Paige and Tarjan's algorithm [27] for strong bisimulation instead of Kanellakis and Smolka's, or using approaches for symbolic bisimulations [31].

Notions of bisimulations for FuTSs, ULTraSs, and variations thereof have been shown to be as expressive as those for WLTSs [23, 25, 28]. The extension of these results to weak bisimulation will automatically yield decision procedures for FuTSs and ULTraSs based on our algorithm.

Bibliography

[1] L. Aceto, A. Ingólfsdóttir, and J. Sack. Resource bisimilarity and graded bisimilarity coincide. *Inf. Proc. Letters*, 111(2):68–76, 2010.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.

[3] C. Baier. On algorithmic verification methods for probabilistic systems. Universität Mannheim, 1998.

[4] C. Baier and H. Hermanns. Weak bisimulation for fully probabilistic processes. In *Proc. CAV*, pages 119–130, 1997.

[5] C. Baier, B. Engelen, and M. E. Majster-Cederbaum. Deciding bisimilarity and similarity for probabilistic processes. *JCSS*, 60(1):187–231, 2000.

[6] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *TCS*, 202 (1-2):1–54, 1998.

[7] M. Bernardo, R. De Nicola, and M. Loreti. A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences. *Inf. Comput.*, 225:29–82, 2013.

[8] T. Brengos and M. Peressotti. A uniform framework for timed automata. In *Proc. CONCUR*, volume 59 of *LIPIcs*, pages 26:1–26:15, 2016.

[9] T. Brengos, M. Miculan, and M. Peressotti. Behavioural equivalences for coalgebras with unobservable moves. *JLAMP*, 84(6):826–852, 2015.

[10] R. De Nicola, D. Latella, M. Loreti, and M. Massink. A uniform definition of stochastic process calculi. *ACM Computing Surveys*, 46(1):5, 2013.

[11] D. Grigoriev and V. V. Podolskii. Complexity of tropical and min-plus linear prevarieties. *CoRR*, abs/1204.4578, 2012.

[12] H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *TCS*, 274(1-2):43–87, 2002.

[13] J. Hillston. *A compositional approach to performance modelling*. CUP, 1996.

[14] P. C. Kanellakis and S. A. Smolka. Ccs expressions, finite state processes, and three problems of equivalence. *Inf. Comput.*, 86(1):43–68, 1990.

[15] B. Klin. Structural operational semantics for weighted transition systems. In J. Palsberg, editor, *Semantics and Algebraic Specification*, volume 5700 of *LNCS*, pages 121–139. Springer, 2009.

[16] B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In *Proc. FoSSaCS*, volume 4962 of *LNCS*, pages 428–442, 2008.

[17] B. Klin and V. Sassone. Structural operational semantics for stochastic and weighted transition systems. *Inf. Comput.*, 227:58–83, 2013.

[18] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

[19] D. Latella, M. Massink, and E. P. de Vink. Bisimulation of labelled state-to-function transition systems coalgebraically. *LMCS*, 11(4), 2015.

[20] G. L. Litvinov, V. P. Maslov, A. Y. Rodionov, and A. N. Sobolevski. Universal algorithms, mathematics of semirings and parallel computations. In A. N. Gorban and D. Roose, editors, *Coping with Complexity: Model Reduction and Data Analysis*, pages 63–89. Springer, 2011.

[21] M. Miculan and M. Peressotti. Weak bisimulations for labelled transition systems weighted over semirings. *CoRR*, abs/1310.4106, 2013.
[22] M. Miculan and M. Peressotti. GSOS for non-deterministic processes with quantitative aspects. In N. Bertrand and L. Bortolussi, editors, *Proc. QAPL*, volume 154 of *EPTCS*, pages 17–33, 2014.
[23] M. Miculan and M. Peressotti. On the bisimulation hierarchy of state-to-function transition systems. In *ICTCS*, volume 1720 of *CEUR Workshop Proceedings*, pages 88–102. CEUR-WS.org, 2016.
[24] M. Miculan and M. Peressotti. Structural operational semantics for non-deterministic processes with quantitative aspects. *TCS*, 655 Part B:135–154, 2016.
[25] M. Miculan and M. Peressotti. Reductions for transition systems at work: Deriving a logical characterization of quantitative bisimulation. *CoRR*, abs/1704.07181, 2017.
[26] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
[27] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
[28] M. Peressotti. On the trade-off between labels and weights in quantitative bisimulation. *CoRR*, abs/1705.06439, 2017.
[29] C. Priami. Stochastic pi-calculus. *Comput. J.*, 38(7):578–589, 1995.
[30] R. J. van Glabbeek. The linear time—branching time spectrum. In *Proc. CONCUR*, pages 278–297, 1990.
[31] R. Wimmer, M. Herbstritt, H. Hermanns, K. Strampp, and B. Becker. Sigref–a symbolic bisimulation tool box. In *Proc. ATVA*, pages 477–492. Springer, 2006.